THE HECKMAN BINDERY, INC.
North Manchester, Indiana

KRL

H or V	JUST	FONT	SLOT	TITLE
H	CC	1W	22	BBBR
			21	FACULTY
			20	WORKING
			19	PAPER
H	CC	1W	8	1989
			7	NO.1540-1554
H	CC	1W		330
				B385<'CV">
				no.1540-1554
				cop.2
H	CC	7W		<IMPRINT>
				U. of ILL.
				LIBRARY
				URBANA

BINDING COPY

PERIODICAL ☐ CUSTOM ☐ STANDARD ☐ ECONOMY ☐ THESIS | NO. VOLS. THIS TITLE | LEAD ATTACH.

BOOK ☐ CUSTOM ☐ MUSIC ☐ ECONOMY | AUTH 1ST ☐ | COLOR

ACCOUNT | LIBRARY | NEW | RUB OR SAMPLE | TITLE I.D | FOIL | MATERIAL

56572 001 | 6632 | WHI | 488

ACCOUNT NAME

UNIV OF ILLINOIS

ACCOUNT INTERNAL I.D. | ISSN.

B01912400

I.D. #2 | NOTES | BINDING FREQUENCY | WHEEL | SYS. I.D.

1 | 3 | 39256

STX3

COLLATING

35

ADDITIONAL INSTRUCTIONS

Dept=STX3 Lot=#20 Item=142 PNM=1877
1CR2ST3CR MARK BY # B4 91

SEP SHEETS	PTS. BD. PAPER	TAPE STUBS	CLOTH EXT	FILLER GUM	FILLER STUB

POCKETS: PAPER | BUCK | CLOTH

SPECIAL PREP | LEAF ATTACH

INSERT MAT | ACCOUNT LOT NO: #20 | JOB NO: H0363

PRODUCT TYPE: 1 | ACCOUNT PIECE NO: 142 | PIECE NO: 89

HEIGHT | GROUP CARD | VOL THIS TITLE

COVER SIZE: X

0012475

BEBR
FACULTY WORKING
PAPER NO. 89-1549

A Bicriterion Scheduling Problem in a General Flexible Manufacturing System

N. Raman

College of Commerce and Business Administration
Bureau of Economic and Business Research
University of Illinois Urbana-Champaign

FACULTY WORKING PAPER NO. 89-1549

College of Commerce and Business Administration

University of Illinois at Urbana-Champaign

April 1989

A Bicriterion Scheduling Problem in a General Flexible Manufacturing System Minimize Tardiness Related Costs

N. Raman, Assistant Professor
Department of Business Administration

A Bicriterion Scheduling Problem in a General Flexible Manufacturing System

N. Raman
Department of Business Administration
University of Illinois
1206 South Sixth Street
Champaign, IL 61820

ABSTRACT

Surveys of industrial scheduling practices note that operating managers judge the effectiveness of a shop schedule on several criteria. Yet, much of the existing research on scheduling has considered only single-objective problems. In this paper, we address a bicriterion problem for a flexible manufacturing system which produces parts to specific orders. The primary objective of the scheduling problem is the minimization of total job tardiness, and the secondary objective is to maximize the sum of job release times.

The significance of the primary objective for any make-to-order system is obvious. The secondary objective is important in systems operating in a just-in-time environment with its emphasis on minimizing in-process inventories. We present an integer programming formulation of this problem, and construct a heuristic hierarchical solution method which requires decomposing the original problem into two subproblems which address the two objectives independently. Computational studies show the effectiveness of the suggested solution procedure under various test scenarios.

1. INTRODUCTION

This paper addresses the Shop Scheduling Problem (SSP) in a flexible manufacturing system (FMS) with random material flows. The SSP is a dual-objective problem; the objective of primary concern is to minimize total job tardiness, while the secondary objective is to maximize the sum of job release times. The requirements of these two objectives are antithetical. While minimizing total tardiness is a regular measure, maximizing the sum of release times is non-regular.

In considering these two objectives, and by ordering them lexicographically, we associate two characteristics with the make-to-order system addressed in this study: i) It operates in a just-in-time (JIT) manufacture environment with emphasis on its short- and long-term benefits; and ii) in the short-term, minimizing the cost of tardiness is of primary importance. These characteristics are now discussed in some detail.

The importance of the primary objective for any make-to-order system is obvious. The secondary objective (and most non-regular measures in general) is consistent with the JIT manufacture notion that long-term benefits due to productivity and quality improvements, and short-term savings, such as reductions in work-in-process (WIP) and finished goods inventory carrying charges, can outweigh the cost of keeping resources idle for a certain

length of time. In addition, when part processing times are large and the due dates of some orders extend very much into the future, the system needs to be buffered against the quantity and timing uncertainties associated with such orders. In an FMS in particular, because of the significant costs of tool preparation and part programming, it is desirable to defer the commitment of resources to these orders to a later point in time when such uncertainties are reduced.

The feasibility of deferring part production in JIT systems, while simultaneously ensuring that their due dates are met on most occasions, stems, at least partly, from the availability of surplus production capacity. However, in spite of such extra capacity, there may be instances in which some jobs arriving at the GFMS need to be taken up urgently. In such cases, the operating manager is faced with the objective of minimizing due date based penalties (such as tardiness) associated with the urgent jobs, while retaining the objective of deferring the release times of the other jobs. [Because of interaction effects, it may not always be possible a priori to identify the urgent jobs.]

In treating the two objectives lexicographically, we make a distinction between the short- and the long-term priorities of the manager. We assume that, for the system addressed in this study, the <u>short-term</u> benefits of JIT manufacture are

significantly less than the tardiness-related costs. Therefore, in the event of conflict between the two objectives, minimizing tardiness is given higher priority. When excess capacity is available, short-term conflicts between the two antithetical objectives are avoided on most occasions.

Note that deferring job release times undermines the system's ability to meet the due dates of jobs arriving in the future. Even in the presence of adequate production capacity, it can lead to a small deterioration in tardiness values in a dynamic system. Therefore, if only the short-term costs and benefits are considered, it is never optimal to defer job release times. [In fact, it will probably be suboptimal to do so.] In the long term, however, benefits due to improvements in productivity and quality can compensate for small increases in tardiness. We make another observation here. Keeping resources idle for production purposes does not imply that they will remain unutilized. In practice, they will be put to alternative uses such as part program testing, prototype development, etc. While these auxiliary activites are important for smooth ongoing production in an FMS, they are usually scheduled around the schedules developed for jobs with specific orders.

This paper is organized as follows. We review the prior research on related problems in Section 2. An integer programming formulation of this problem is presented in Section 3. In

Section 4, we discuss the heuristic solution approach which is based on a hierarchical decomposition of the problem into independent subproblems. Our computational experience with this approach is described in Section 5. We conclude in Section 6 with a discussion of the experimental results. The notation used in this chapter is given in Appendix 1.

2. BACKGROUND

The Shop Scheduling Problem has not been addressed directly by any researcher in the past. However, the two objectives have been studied individually. While the total tardiness problem has atttracted extensive attention, the research on this problem has primarily focused on the use of priority dispatching rules in dynamic systems. [See, for example, Carroll (1965), Baker and Bertrand (1982), Kanet and Hayya (1982), Baker and Kanet (1983), Baker (1984) and Vepsalainen and Morton (1987)]. In a recent study, Raman, Talbot and Rachamadugu (1989) develop an implicit enumeration approach as well as a decomposition-based heuristic for solving the static version of this problem.

The secondary objective considered in this paper is closely related to the earliness problem studied by Sidney (1977), Lakshminarayan et al. (1978), Chand and Schneeberger (1985, 1986), and Rachamadugu (1986). Sidney (1977) addresses a problem in which jobs incur both lateness and earliness penalties; the

objective is to minimize the maximum of these penalties over all jobs. Under certain restrictive assumptions, Sidney develops the properties of the optimal ordering of jobs and presents an $O(N^2)$ algorithm for solving the problem optimally. Lakshminarayan et al. (1978) present an O(N log N) algorithm for solving the same problem.

Chand and Schneeberger (1985, 1986) consider the single machine weighted earliness problem. They show that this problem is antithetical to the problem of minimizing the total weighted job completion times subject to no late jobs. They propose a dynamic programming-based algorithm for solving the earliness problem exactly. They also construct a heuristic solution method which requires a modification of the procedure suggested by Smith (1956). Rachamadugu (1986) derives valid lower bounds for the weighted earliness problem by splitting jobs. He formulates a Lagrangean problem by relaxing the constraints on early job completion. The Lagrangean variables are determined by a multiplier adjustment procedure.

As we will show later in Section 3, the secondary objective of maximizing the sum of job release times is equivalent to minimizing the sum of job waiting times. The latter problem is considered by Ahmadi and Bagchi (1987) in the context of a two-machine flow shop. They present several dominance properties and a branch and bound method for solving this problem optimally. In

this method, the lower bound at any node is derived by using a modification of Johnson's procedure for minimizing makespan. The upper bound at any node is determined by using the Shortest Remaining Processing Time heuristic for solving the antithetical completion time problem subject to non-zero ready times and a common due date. However, when all jobs do not have the same due date, Ahmadi and Bagchi restrict their search to permutation schedules only. They also outline a heuristic procedure for the multiple (≥3) machine flow shop.

This study extends the previous research to the case of an FMS. Because of the random material flows, the FMS problem is considerably more complicated than the single machine and flow shop problems. To solve any problem of reasonable size, therefore, we have to necessarily adopt a heuristic solution approach. The details of the proposed decomposition-based heuristic method are discussed next.

3. PROBLEM FORMULATION

An integer programming formulation of SSP is given below:

$$\text{Maximize} \quad \sum_j r_j \qquad (1)$$

subject to

$$\sum_j T_j = z_1 \text{ . where } z_1 = \min \sum_j T_j \qquad (2)$$

$$\sum_{t} x_{tjk} = 1, \text{ for each } j, k \qquad (3)$$

$$\sum_{j}\sum_{k}\sum_{q=t}^{t+p_{jk}-1} R_{jkm}\, X_{qjk} \le 1, \text{ for each } m, t \qquad (4)$$

$$\sum_{j} (t - p_{j1})\, x_{tj1} \ge \begin{cases} r_j, & \text{if } j = 1 \\ \sum_{t} t\, x_{tjk}, & \text{otherwise} \end{cases} \qquad (5)$$

$$\text{for each } j, k, \text{ and } (k,1) \in S_j$$

$$\sum_{t} t\, X_{tjk} + E_j - T_j = d_j, \text{ for each } j, \text{ and } k = N_j \qquad (6)$$

$$x_{tjk} \in \{0,1\}, \text{ for each } j, k, t; \quad r_j, E_j, T_j \ge 0, \text{ integer for each } j \qquad (7)$$

In this formulation, Equation (1) refers to the secondary objective. Constraints (2) specify the primary objective. Constraints (3) ensure that each operation is completed exactly once. Constraints (4) require that each machine processes only one operation in any time period. Constraints (5) ensure that any operation can be scheduled only after its predecessor operation is completed, and the first operation of any job is started only after the job is released. Constraints (6) define tardiness while Equation (7) specifies the nature of the variables.

SSP remains NP-complete. We propose to simplify this problem by decomposing the set of available jobs J. J can be treated as the union of two mutually exclusive and collectively exhaustive sets J_1, which consists of urgent jobs, and J_2 which comprises those jobs which can be deferred. If we can effectively partition J into J_1 and J_2, then the primary objective of SSP can be seen to be the minimization of total tardiness of jobs in J_1, and the secondary objective as the maximization of the sum of release times of jobs in J_2, subject to the requirement that these jobs be completed by their respective due dates. Note that this decomposition ignores the interaction between the primary and the secondary objectives for jobs in J_1. In effect, if J_1 is not empty, we assume that there are no alternative optimal schedules for the minimum tardiness problem with different values of the sum of job release times. The earlier formulation of SSP can then be modified as below.

$$\text{Maximize} \quad z_2 = \sum_j r_j \qquad (1)'$$

subject to

$$\sum_j T_j = z_1, \text{ where } z_1 = \min \sum_{j \in J_1} T_j \qquad (2)'$$

$$\sum_t x_{tjk} = 1, \text{ for each } j, k \qquad (3)$$

$$\sum_j \sum_k \sum_{q=t}^{t+p_{jk}-1} R_{jkm}\, x_{qjk} \leq 1, \text{ for each } m, t \qquad (4)$$

$$\sum_t (t - p_{j1})\, x_{tj1} \geq \begin{cases} r_j, & \text{if } l=1 \\ \sum_t t\, x_{tjk}, & \text{otherwise} \end{cases} \qquad (5)$$

for each j, k, and $(k,l) \in S_j$

$$\sum_t t\, x_{tjk} + E_j - T_j = d_j, \text{ for each } j, \text{ and } k = N_j \qquad (6)$$

$$x_{tjk} \in \{0,1\}, \text{ for each } j,k,t; \quad r_j, E_j, T_j \geq 0, \text{ integer, for each } j \qquad (7)$$

$$T_j = 0, \text{ for each } j \in J_2 \qquad (8)$$

Equation (8) in the above formulation ensures that jobs in J are completed by their due dates. This formulation shows that two combinatorial problems are embedded within SSP. Equations (1)′, (3), (4), (5), (7), and (8) constitute the Input Scheduling Problem (ISP) of maximizing the sum of the release times of jobs in J_2 subject to their timely completion. This problem has been shown to be NP-complete even when $M=1$ [see Chand and Schneeberger (1986)]. Equations (2)′, (3), (4), (5), (6), and (7) define the Multiple-Machine Tardiness Problem (MTP) of minimizing the total tardiness of jobs in J_1.

For given job due dates, maximizing $\sum_{j \in J_2} r_j$ is equivalent to minimizing $\sum_{j \in J_2} (d_j - r_j)$. Also,

$$\sum_{j\varepsilon J_2} (d_j - r_j) = \sum_{j\varepsilon J_2} \sum_{i=1}^{N_j+1} (c_{ji} - c_{j,i-1})$$

where c_{ji} denotes to completion time of operation i in job j, $c_{j,Nj+1} = d_j$, and $c_{j0} = r_j$. If we denote the waiting time of job j after operation i by W_{ji}, then

$$c_{ji} - c_{j,i-1} = W_{j,i-1} + p_{ji}$$

Also, in an optimal schedule, a job will be released immediately before the start of its first operation. Therefore,

$W_{j0} = 0$, and

$$c_{j1} - c_{j0} = c_{j1} - r_j = p_{j1}$$

Hence, in an optimal schedule,

$$\sum_{j\varepsilon J_2} (d_j - r_j) = \sum_{j\varepsilon J_2} \sum_{i=1}^{N_j} W_{ji} + \sum_{j\varepsilon J_2} \sum_{i=1}^{N_j} p_{ji}$$

Since the second term on the right hand side of the above equation is a constant, the objective of minimizing $\sum_{j\varepsilon J_2} (d_j - r_j)$ is equivalent to minimizing the sum of job waiting times. [Note that the definition of waiting time used above includes earliness as well.] For the single machine case, this objective reduces to minimizing total earliness.

4. SOLUTION APPROACH

The modified formulation given in the previous section motivates a hierarchical solution procedure for SSP. At the higher level, we address the problem of determining J into J_1 and J_2. The resulting subproblems, MTP for jobs in J_1 and ISP for jobs in J_2, are then solved at the lower level. These three problems are now discussed.

4.1 Determination of J_1 and J_2

To generate J_1 and J_2, we solve an aggregate problem in which the entire FMS is treated as a single machine which processes batches of jobs. A batch consists of jobs which are to be processed simultaneously. The sequence of jobs within a batch maximizes the sum of job release times subject to their timely completion.

Based on this sequence, we determine the processing time and the due date of each batch. The objective of the aggregate problem is to determine whether a feasible sequence of these batches exists in which all batches are completed before their due dates. If there is no such sequence, batches are combined in order to increase the likelihood of feasibility. The suggested procedure for combining batches terminates in one of the following two ways: i) a feasible sequence of batches is obtained, or

ii) there is exactly one batch which is tardy.

Jobs which belong to the tardy batch, if any, are assigned to J_1. These jobs are resequenced according to the solution to MTP which is discussed in Raman, Talbot and Rachamadugu (1989). The remaining jobs constitute J_2 and define the Input Scheduling Problem which is discussed in Section 4.3.

Partitioning J into J_1 and J_2, therefore, requires three steps: i) formation of batches, ii) determination of batch processing times and batch due dates, and iii) test for feasibility. These steps are now discussed.

Formation of Job Batches

At the aggregate level, it is meaningful to assume that jobs with similar due dates will be processed together. If no interactions exist, a given job j can be released into the system at time $r_j = d_j - p_j$, to maximize its release time while ensuring that it is completed by its due date. However, when many jobs have similar due dates, job interactions may force one or more jobs to be released earlier.

The objective of the batching problem is to group jobs of similar due dates together. We propose a cluster-analytic solution approach in which the similarity of two jobs is measured in terms of the squared difference of their due dates. Since the number of batches to be formed is not known a priori, it should be

determined as part of the procedure for assigning jobs to batches.

Several methods for clustering a given set of objects have been suggested [see, for example, Cormack (1971) for a classification of these procedures, and Anderberg (1973), Hartigan (1975) and Everitt (1974) for their description]. Computational studies of Everitt (1974) and Klastorin (1985) indicate that most methods perform reasonably well although they may yield different clusters. Both Everitt and Klastorin suggest that several methods should be tried for the same data set, and the dominant partition should be retained. However, it is difficult to do so in view of the computational burden involved. Also, it should be noted that the solution to the batching problem provides only the initial composition of batches. Unless job due dates are very loose and machine utilizations are very low, it is quite likely that these batches will be redefined during the test of feasibility (which is discussed later in this section. From this perspective, this test itself can be viewed as a clustering procedure). Based on this consideration, the approach of using several clustering procedures in order to obtain the best partition of batches is not particularly attractive.

Everitt's (1974) comparative study of several procedures showed the superiority of the single-linkage hierarchical agglomerative approach for forming clusters. In this method, starting from the

lowest level of the hierarchy at which each batch consists of only one job, batches of increasing cardinality are formed by merging two batches at a time. At a given step, the batches selected for merging are those with the minimum squared difference of the mean due dates. The algorithm terminates when all jobs are merged into a single batch. From the batches generated in this manner, the best set of batches is selected by the set-partitioning procedure suggested by Klastorin (1982) which is based on maximizing the measure of "expected distinctiveness". The batches formed in the hierarchical manner are treated as the vertices of a directed arborescent network. The problem of determining the best partition reduces to finding the maximum weight cut-set in this network which can be done in O(N) steps.

- Determination of Batch Parameters

For the given batches, we next determine the processing times and the due dates. In order to do so, we need to determine the sequence of jobs within each batch. Consistent with the previous discussion, this sequence should maximize the sum of job release times subject to the requirement that they should be completed by their due dates. [A time-axis shift is required if one or more jobs are already late.] We call this problem the job sequencing problem (JSP).

The JSP can be viewed as the inversion of the completion time problem (CTP) in which the objective is to minimize the total completion times of jobs with ready times $r'_j = d_{max} - d_j$, where d_{max} is the maximum of the due dates of jobs within the given batch.

To see the equivalence between these two problems consider the example shown in Figure 1. In this 3-job example, we have

$$d_3 < d_2 < d_1 = d_{max}$$

for the JSP. The corresponding instance of CTP is constructed by defining the following job ready times:

$$r'_1 = 0,\ r'_2 = d_1 - d_2,\ \text{and}\ r'_3 = d_1 - d_3$$

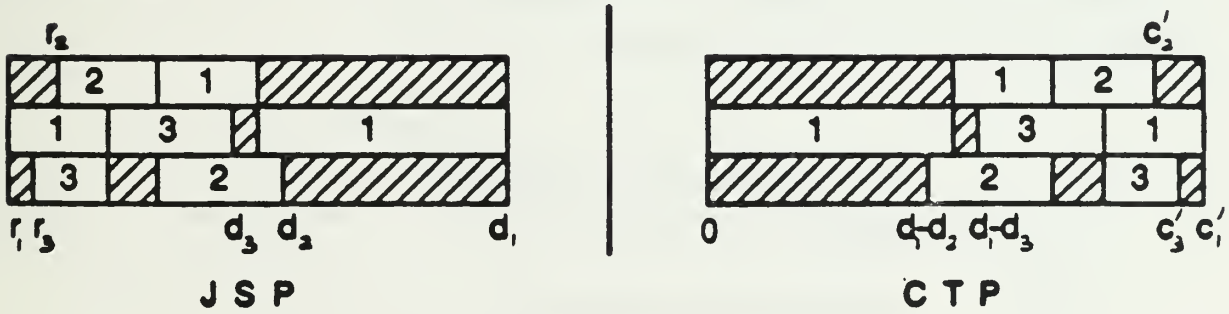

Figure 1 - Equivalence of JSP and CTP

Denoting the variables in CTP with primes, it can be seen that, in general,

$$c'_{max} = c_{max},\ \text{and}\ c'_j = d_{max} - r_j .$$

Therefore, maximizing $\Sigma_j r_j$ in JSP is equivalent to minimizing $\Sigma_j c'_j$ in CTP. The desired solution to JSP, in terms of job ready times, can then be obtained by first solving CTP to determine job completion times c'_1, c'_2, and c'_3, and then using the following relationship:

$$r_1 = d_1 - c'_1 = 0,\ r_2 = d_1 - c'_2,\ \text{and}\ r_3 = d_1 - c'_3.$$

However, CTP is itself a hard problem. While this problem has been addressed for the single machine case [see, for example, Hariri and Potts (1983)], to our knowledge it has not been extended to multiple machines. We propose a heuristic procedure which is a modification of Smith's (1956) algorithm for a single machine problem, and which generates a non-delay schedule. The procedure is given below. However, first we give the additional notation used in the procedure.

O_j	The schedulable operation of job j
$r(O_j)$	Ready time of O_j
$p(O_j)$	Processing time of O_j
$\mu(O_j)$	The machine on which O_j is to be processed
Γ	The set of schedulable jobs

Algorithm:

Step 1 Initialize:

Γ = the set of all jobs in batch b;

count (m) = 0, t(m) = 0, for m = 1,.., M.

Go to Step 2.

Step 2 Determine the next operation assignment time t:

$\sigma = \{O_j \mid j \in \Gamma\}$, $M(\sigma) = \{m \mid \mu(O_j) = m\}$;

$$t'(m) = \begin{cases} \min\limits_{O_j \in \sigma} \{r(O_j) \mid \mu(O_j) = m\}, & \text{if } m \in M(\sigma) \\ 0, & \text{otherwise;} \end{cases}$$

$t(m) = \max[t(m), t'(m)]$;

$m^* = \arg\min\limits_{m} \{t(m)\}$; $t = t(m^*)$.

Go to Step 3.

Step 3 Determine the next operation to be assigned:

Find $i = \arg\min\limits_{j \in \sigma'} \{p(O_j)\}$

where $\sigma' = \{j \mid O_j \in \sigma, \mu(O_j) = m^*\}$.

Go to Step 4.

Step 4 Update machine availability time and σ :

count(m*) = count(m*) + 1;

sequence O_j in position count(m*) on m*

$t(m^*) = t(m^*) + p(O_j)$;

if O_j is the last operation of job i, then

$\Gamma = \Gamma \backslash i$,

else, O_j = successor operation of O_j;

$r(O_j) = t(m^*)$.

Go to Step 5.

Step 5 Determine whether all operations are assigned:

If $\Gamma = 0$, stop; else, go to Step 2.

This procedure generates a non-delay schedule by considering only those jobs which are available at a given machine when the scheduling decision is to be made at that machine. Ties between competing jobs are broken in favor of the job with the shortest imminent operation time.

The order of operations generated for CTP is reversed to obtain the desired sequence of jobs within the batch for JSP. Based on this sequence, the batch processing time and the batch due date are determined as follows.

The processing time P_b of batch b equals its makespan. We define the due date D_b of batch b as

$$D_b = \max_{j \in b} \{d_j\}$$

Note that the construction of the sequence of jobs within a batch ensures that the job which is due last is completed last, and other jobs are completed on or before their due dates. Therefore, D_b provides a valid representation of the batch due date in the sense that all jobs are completed by their due date if and only if the batch that they belong to is completed by its batch due date.

Test for Feasibility

The procedure used for testing feasibility is outlined below.

Step 1 Initialize:

s = 1;

B = total number of batches currently available.

Go to Step 2.

Step 2 Generate an EDD sequence of the remaining batches:

Starting from the batch in position s, generate an EDD sequence of all batches.

Go to Step 4.

Step 3 Determine the next batch to be investigated:

s = s + 1;

if s = B + 1, go to step 6;

else, go to Step 4.

Step 4 Determine if this batch is tardy:

If $C_{[s]} > D_{[s]}$, go to Step 5;

else, go to Step 3.

Step 5 Combine batches, if possible:

If s = 1, go to Step 3;

else, combine [s] with [s-1];

s = s - 1; B = B - 1;

redefine $P_{[s-1]}$ and $D_{[s-1]}$.

Go to Step 2.

Step 6 Determine J_1 and J_2:

$J_1 = \{[1] | [1] \text{ is late}\}$, and

$J_2 = J \backslash J_1$

This procedure first arranges all batches in the EDD sequence. Because EDD minimizes maximum lateness, a feasible solution exists if and only if there are no late batches in this sequence, which is scanned from front to back. If no late batches are found, this procedure terminates. Otherwise, the first late batch found is merged with its immediate predecessor, and jobs belonging to these two batches are resequenced following the procedure outlined in Section 4.2. This step is likely to

increase the likelihood of feasibility as shown by the following result.

THEOREM 1: The total tardiness of any sequence is not increased if a late batch is merged with the batch immediately preceding it.

PROOF: Refer to Appendix 2.

Because at each stage we combine the late batch, if any, with its immediate predecessor, this procedure will terminate in at most B - 1 steps in one of the following two ways: i) A sequence is found in which all batches are early, or ii) a sequence is obtained in which the first batch is tardy and other batches, if any, are completed by their due dates.

Jobs which belong to the tardy batch are assigned to J_1. These jobs are resequenced in accordance with the solution to MTP. Note that this resequencing may alter the makespan of the first batch. In such a case, the 3-step procedure discussed above is repeated after redefining the processing time of this batch. The remaining jobs are assigned to J_2. The scheduling of jobs in J_1 and J_2 is now discussed.

4.2 The Multiple-Machine Tardiness Problem

Jobs in set J_1 are urgent and they constitute the subproblem pertaining to the primary objective of minimizing total tardiness. This problem is identical to the static MTP considered in Raman, Talbot and Rachamadugu (1989), and the same solution approaches can now be applied to sequence jobs in J_1.

4.3 The Input Scheduling Problem

The objective of ISP is to maximize the sum of release times of jobs in J_2 subject to their timely completion. We propose solving this problem by using the Modified Smith Heuristic (MSH) procedure described in Section 4.1.

In general, when job due dates are widely dispersed, MSH will yield several "blocks" of jobs. Each block consists of jobs which are processed together. Some or all of these blocks may correspond to the job batches formed as part of the aggregate problem considered in Section 4.1.

We now discuss the experimental investigation of the proposed solution approach.

5. EXPERIMENTAL STUDY

The experimental investigation considered two objectives which were addressed individually by the two sets of experiments conducted. The first set of experiments evaluated the merit of addressing the dual-objective problem over the single-objective mean tardiness problem. While it is clear that a solution to the former should provide better values of the sum of job release times, the purpose of this set of experiments was to examine the margin of possible improvement.

The objective of the second set of experiments was to judge the effectiveness of MSH procedure with respect to known upper bounds. From the due date constraints, we have

$$r_j + p_j \leq d_j \text{ , for } j \in J_2$$

Hence,

$$\sum_{j \in J_2} r_j \leq \sum_{j \in J_2} d_j - \sum_{j \in J_2} p_j = UB$$

However, UB provides a weak upper bound. Ahmadi and Bagchi (1987) derive an upper bound for a similar problem for a two-machine flow shop in which all job due dates are equal. [They derive the upper bound also for unequal job due dates; it is, however, valid only for permutation schedules which are not dominant for non-regular measures.] To our knowledge, this is the only other upper bound available currently for multiple

machine problems addressing similar objectives. We will denote this upper bound by UB_o.

The second set of experiments, consequently, addressed a 2-machine flow shop with equal job due dates and compared the solution value yielded by MSH with respect to UB_o.

5.1 Experimental Design

The first set of experiments considered a 5-machine system with 25 and 50 jobs. The number of operations within each job varied between 1 and 5. Each job followed a random machine visitation sequence though successive operations of a given job were processed in different machines. Operation processing times varied uniformly in the interval (12,87).

The due date of d_j of a given job j was determined by

$$d_j = F \left(\sum_j p_j \right)$$

F was sampled from a uniform distribution in the interval (F - RF/2, F + RF/2). F and R respectively control the tightness and the variability of job due dates. In this study, six values of F - 0.1, 0.2, 0.3, 0.4, 0.5, and 0.6, and two values of R-0.5, and 1.5, were used to provide twelve combinations of due date tightness and variability.

Ten instances of each problem scenario were randomly generated. Each instance was solved using two approaches. The first approach employed the sequential solution procedure designed to consider the two objectives lexicographically as given in Section 4. The second approach considered only the primary objective of minimizing total tardiness. The solution values with respect to total tardiness and the sum of job release times obtained under both approaches were recorded and averaged over the ten problem instances for reporting purposes. In order to restrict the computational costs within reasonable limits, the Modified Operation Due Date (MOD) rule [see, for example, Baker (1984)] was used for solving the mean tardiness problem under both approaches. Hereafter, we will refer to these two approaches as SSP/MOD and MOD respectively. In total, the first set of experiments considered 240 problems.

The second set considered a 2-machine system with 25, 50 and 100 jobs. Each job had two operations, one at each machine, and it visited machine 1 first. Operation processing times were selected from a uniform distribution in the interval (12,87). All jobs had the same due date d which was determined by

$$d = F \left(\sum_{j} p_j \right).$$

Five values of F - 0.6, 0.7, 0.8, 0.9, and 1.0 were used to generate increasingly loose due dates. [As reported in Ahmadi

and Bagchi's (1987) study as well, we found that F < 0.6 led to due date infeasibility in many cases.]

Ten instances of each problem were generated. For each instance, the ratio of the sum of job release times obtained from SSP to the upper bound derived by using Ahmadi and Bagchi's approach was recorded. The results report the average as well as the minimum and the maximum values of these ratios over the ten problem instances. In all, the second set of experiments considered 150 problems.

5.2 Experimental Results

Tables 1 through 3 give the results of the experiments conducted. The performance of SSP/MOD and MOD with respect to total tardiness and the sum of job release times is shown in Table 1. For better comparison, the reported total tardiness values are normalized with respect to the sum of job processing times for better comparison. Similarly, the sum of job release times is normalized with respect to the sum of job due dates. In Table 1, z_1, and z_2 denote normalized total tardiness and normalized sum of release times respectively.

Table 2 reports the ratio of the sum of job release times obtained under SSP/MOD to the upper bound UB discussed in the

beginning of Section 5 for the eight different values of F and R under which all jobs were completed on time.

Table 3 presents the results of the second set of experiments. The values reported are the ratios of the SSP/MOD solution value to the upper bound UB_0 given by Ahmadi and Bagchi´s procedure.

TABLE 1

COMPARISON OF SSP/MOD AND MOD

Number of Jobs = 25

R = 0.5

F	(z_1, z_2)		$\frac{z_2 (SSP/MOD)}{z_2 (MOD)}$
	SSP/MOD	MOD	
0.1	(0.559, 0.646)	(0.560, 0.646)	1.00
0.2	(0.030, 0.336)	(0.031, 0.308)	1.09
0.3	(0.000, 0.711)	(0.000, 0.205)	3.47
0.4	(0.000, 0.813)	(0.000, 0.154)	5.28
0.5	(0.000, 0.871)	(0.000, 0.123)	7.08
0.6	(0.000, 0.903)	(0.000, 0.102)	8.85

R = 1.5

F	(z_1, z_2)		$\frac{z_2 (SSP/MOD)}{z_2 (MOD)}$
	SSP/MOD	MOD	
0.1	(0.526, 0.660)	(0.532, 0.658)	1.00
0.2	(0.038, 0.434)	(0.042, 0.343)	1.26
0.3	(0.001, 0.762)	(0.002, 0.230)	3.31
0.4	(0.000, 0.879)	(0.000, 0.173)	5.08
0.5	(0.000, 0.906)	(0.000. 0.138)	6.56
0.6	(0.000, 0.924)	(0.000, 0.115)	8.03

TABLE 1 (CONTINUED)

COMPARISON OF SSP/MOD AND MOD

Number of Jobs = 50

R = 0.5

F	(z_1, z_2)		$\frac{z_2 (SSP/MOD)}{z_2 (MOD)}$
	SSP/MOD	MOD	
0.1	(0.430, 0.646)	(0.431, 0.646)	1.00
0.2	(0.004, 0.454)	(0.004, 0.315)	1.44
0.3	(0.000, 0.793)	(0.000, 0.210)	3.78
0.4	(0.000, 0.881)	(0.000, 0.157)	5.61
0.5	(0.000, 0.923)	(0.000, 0.126)	7.32
0.6	(0.000, 0.945)	(0.000, 0.105)	9.00

R = 1.5

F	(z_1, z_2)		$\frac{z_2 (SSP/MOD)}{z_2 (MOD)}$
	SSP/MOD	MOD	
0.1	(0.387, 0.687)	(0.389, 0.683)	1.01
0.2	(0.009, 0.626)	(0.011, 0.347)	1.80
0.3	(0.000, 0.911)	(0.000, 0.230)	3.96
0.4	(0.000, 0.938)	(0.000, 0.173)	5.42
0.5	(0.000, 0.953)	(0.000, 0.138)	6.90
0.6	(0.000, 0.962)	(0.000, 0.115)	8.36

TABLE 2

COMPARISON OF z_2 (SSP/MOD) WITH UB

z_2 (SSP/MOD)/UB

F	NJ = 25		NJ = 50	
	R = 0.5	R = 1.5	R = 0.5	R = 1.5
0.3	0.821	0.880	0.850	0.976
0.4	0.904	0.978	0.927	0.987
0.5	0.947	0.986	0.962	0.993
0.6	0.968	0.991	0.978	0.995

5.3 Analysis of Results

From Table 1, SSP/MOD can be seen to retain the effectiveness of MOD with respect to tardiness while simultaneously yielding better values of the sum of job release times. [The marginal improvement observed in some tardiness values under SSP/MOD is due to the fact that it uses a second tie-breaking rule. When two operations are found to have the same MOD value at the time a scheduling decision is to be made, SSP/MOD favors the operation of the job in the earlier batch. Operationally, this tie-breaking method translates into the EDD rule. However, as seen from the results, the impact of the second tie-breaking rule is very small.]

TABLE 3

COMPARISON OF z_2 (SSP/MOD) WITH UB_0

2-Machine System

Number of Jobs	F	z (SSP/MOD)/UB_0		
		Minimum	Maximum	Average
25	0.6	0.908	0.984	0.945
	0.7	0.928	0.987	0.957
	0.8	0.941	0.990	0.965
	0.9	0.950	0.991	0.970
	1.0	0.957	0.993	0.974
50	0.6	0.912	0.978	0.941
	0.7	0.930	0.983	0.953
	0.8	0.943	0.986	0.962
	0.9	0.951	0.988	0.967
	1.0	0.958	0.990	0.972
100	0.6	0.926	0.960	0.942
	0.7	0.942	0.968	0.954
	0.8	0.952	0.974	0.962
	0.9	0.959	0.978	0.968
	1.0	0.964	0.981	0.972

When due dates are quite tight, e.g., at F = 0.1, there is virtually no difference between SSP/MOD and MOD. However, as due dates become progressively looser, the relative performance of SSP/MOD improves. Even at moderate due date tightness at F = 0.2 and F = 0.3, SSP/MOD can be seen to provide significantly better values of z_2. The relative performance of SSP/MOD improves with an increase in the number of jobs and due date variability.

The dependence of SSP/MOD's (or MSH's) performance on due date variability is brought out in Table 2 as well, especially for the larger problems. Even when F = 0.3, MSH yields release times close to the upper bound when R equals 1.5. This is due to the fact that, with greater dispersion in job due dates, the final schedule contains a larger number of blocks. Consequently, a larger portion of jobs is completed close to their due dates.

It is difficult to judge MSH's performance from Table 2 at low values of F and R because of the fact that UB provides a weak bound. However, from Table 3, it can be seen that MSH performs effectively for these cases as well, at least for the 2-machine system investigated.

6. SUMMARY

This study considers the dual objectives of minimizing total job tardiness and maximizing the sum of job release times in a

lexicographic manner. The proposed solution approach decomposes the set of available jobs into two subsets, thereby generating two subproblems which address the two objectives independently. The first subproblem considers the objective of minimizing total tardiness and it can be solved using the procedure suggested in Raman, Talbot and Rachamadugu (1989). A heuristic solution method is constructed to solve the second subproblem of maximizing the sum of job release times.

The experimental study indicates that significant benefits can result from considering the dual-objective problem instead of only the single-objective total tardiness problem. This result is of interest to an operating manager who is responsible for controlling WIP inventory levels in addition to meeting job due dates effectively. Under the proposed decomposition approach, the quality of the solution to the overall problem is determined individually by the solutions to the total tardiness problem and the job release time problem. In Raman, Talbot and Rachamadugu (1989), we proposed an efficient solution procedure for the former. Experimental results shown in this chapter indicate the effectiveness of MSH for solving the latter problem.

APPENDIX 1

NOTATION FOR THE SHOP SCHEDULING PROBLEM

j	Job index, $j = 1, .., N$
J	Set of available jobs = $\{j\}$
b	Batch index, $b = 1, .., B$
m	Machine index, $m = 1, .., M$
t	Time period, $t = 1, .., T$, where T is the scheduling horizon
d_j	Due date of job j
p_j	Processing time of job j
r_j	Release time of job j
c_j	Completion time of job j
S_j	Set of pairs of adjacent operations in job j
N_j	Number of operations in job j
T_j	Tardiness of job j = $\max (0, c_j - d_j)$
E_j	Earliness of job j = $\max (0, d_j - c_j)$
p_{jk}	Processing time of operation k in job j
D_b	Due date of batch b
P_b	Processing time of batch b
C_b	Completion time of batch b
$[s]$	The batch in position s of a given sequence

x_{tjk}

$$\begin{cases} 1, & \text{if operation k of job j is completed at time t} \\ 0, & \text{otherwise} \end{cases}$$

R_{jkm}

$$\begin{cases} 1, & \text{if operation k of job j requires machine m} \\ 0, & \text{otherwise} \end{cases}$$

APPENDIX 2

PROOF OF THEOREM 1

Let σ be the sequence obtained by ordering batches according to EDD as shown in Figure 2. Let y be the first late batch in σ and let batch x be its immediate predecessor. Let σ′ be the sequence obtained when x and y are combined to yield batch z as shown in Figure 3.

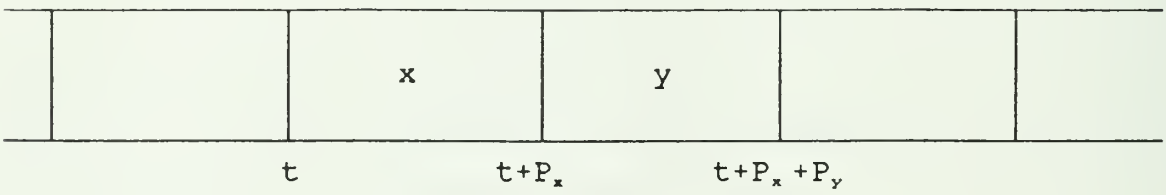

Figure 2 - Sequence σ

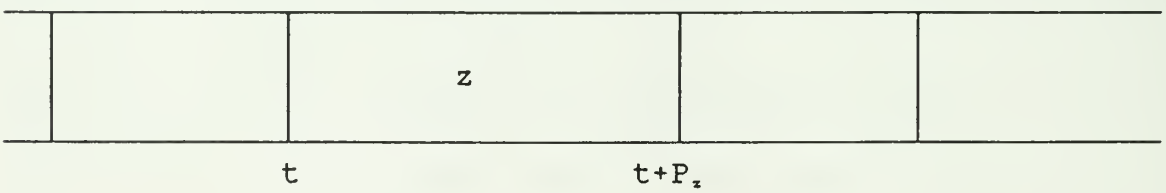

Figure 3 - Sequence σ′

Remark 1: $D_z = D_y$.

Proof: From the definition of batch due dates, we have

$$D_x = \{d_{x^*} \mid x^* = \arg\max_{j \in x} (d_j)\}$$

Similarly, $D_y = \{d_{y^*} \mid y^* = \arg\max_{j \in y} (d_j)\}$

and $$D_z = \{d_{z*} \mid z^* = \arg\max_{j \in z} (d_j)\}$$

Since $D_x \leq D_y$, we have $d_{x*} \leq d_{y*}$. Therefore,

$$D_z = d_{z*} = d_{y*} = D_y$$

Remark 2: $P_z \leq P_x + P_y$

Proof: The procedure of sequencing jobs within a batch ensures that all jobs of batch x will precede all jobs of batch y in the combined batch z. Also, since no idle time is permitted this sequence, those jobs which were in y will complete at the same time or earlier in z. The result stated in Remark 2 follows from this observation.

To prove the theorem, it suffices to show that,

$$t + P_z - D_z \leq t + P_x + P_y - D_y$$

or $$D_z - P_z \geq D_y - (P_x + P_y)$$

This results follows directly from Remarks 1 and 2. This completes the proof.

REFERENCES

Ahmadi, R. H. and U. Bagchi (1987), "Just-in-Time Scheduling in Deadline Constrained Environments", Working Paper, University of Texas, Austin, TX.

Anderberg, M. R. (1973), Cluster Analysis for Applications, Academic Press, New York, NY.

Baker K. R. (1984), "Sequencing Rules and Due date Assignments in a Job Shop", Management Science, Vol. 30, 1093-1104.

Baker, K. R. and J. M. W. Bertrand (1982), "A Dynamic Priority Rule for Sequencing Against Due dates", Journal of Operations Management, Vol. 3, 37-42.

Baker, K. R. and J. J. Kanet (1983), "Job Shop Scheduling with Modified Due dates", Journal of Operations Management, Vol. 4, 11-22.

Carroll, D. C. (1965), "Heuristic Sequencing of Single and Multiple Component Jobs", Ph.D. Dissertation, MIT, Cambridge, MA.

Chand, S. and H. Schneeberger (1985), "A Two-Stage Approximation Method for the Single Machine Weighted Earliness Problem", Working Paper #424, Graduate School of Business Administration, The University of Michigan, Ann Arbor, MI.

Chand, S. and H. Schneeberger (1986), "A Note on the Single Machine Scheduling Problem with Minimum Weighted Completion Time and Maximum Allowable Tardiness", Naval Research Logistics Quarterly, Vol. 33, 551-557.

Conway, R. W. (1965), "Priority Dispatching and Job Lateness in a Job Shop", Journal of Industrial Engineering, Vol. 16, 123-130.

Cormack, R. M. (1971), "A Review of Classification", Journal of the Royal Statistical Society, Series A, Vol. 4, 321-367.

Everitt, B. (1974), Cluster Analysis, Halsted Press, New York, NY.

Hariri, A. M. A. and C. N. Potts (1983), "An Algorithm for Single Machine Sequencing with Release Dates to Minimize Total Weighted Completion Time", Discrete Applied Mathematics, Vol. 5, 99-109.

Hartigan, J. (1975), Clustering Algorithms, John Wiley, New York, NY.

Kanet, J. J. and J. C. Hayya (1982), "Priority Dispatching with Operation Due dates in a Job Shop", Journal of Operations Management, Vol. 2, 155-163.

Klastorin, T. D. (1982), "An Alternative Method for Hospital Partition Determination Using Hierarchical Cluster Analysis", Operations Research, Vol. 30, 1134-1147.

Klastorin, T. D. (1985), "The p-Median Problem for Cluster Analysis:A Comparative Test Using the Mixture Model Approach", Management Science, Vol. 31, 84-95.

Lakshminarayan, S., R. Lakshamanan, R. L. Papineau and R. Rochette (1978), "Optimal Single Machine Scheduling with Earliness and Tardiness Penalties", Operations Research, Vol. 26, 1079-1082.

Posner, M. E. (1985), "Minimizing Weighted Completion Times with Deadlines", Operations Research, Vol. 33, 562-575.

Potts, C. N. and L. N. Van Wassenhove (1982), "A Decomposition Algorithm for the Single Machine Total Tardiness Problem", Operations Research Letters, Vol. 1, 177-181.

Rachamadugu, R. V. (1986), "Bounds for Earliness Problems", Working Paper, University of Michigan, Ann Arbor, MI.

Raman, N., F. B. Talbot and R. V. Rachamadugu (1989),"Scheduling a General Flexible Manufacturing System to Minimize Tardiness Related Costs", BEBR Faculty Working Paper # 89-1548, Univ. of Illinois at Urbana-Champaign, Champaign, IL.

Schonberger, R. J. (1984), Japanese Manufacturing Techniques - Nine Hidden Lessons in Simplicity, The Free Press, New York, NY.

Sidney, J. B. (1977), "Optimal Single machine Scheduling with Earliness and Tardiness Penalties", Operations Research, Vol. 25, 62-29.

Smith, W. E. (1956), "Various Optimizers for Single Stage Production", Naval Research Logistics Quarterly, Vol. 3, 56-66.

Vepsalainen, A. P. J. and T. E. Morton (1985), "Priority Rules for Job Shops with Weighted Tardiness Costs", Management Science, Vol. 33, 1035-1047.